Qualitative modelling of unknown interface behaviour

Lee, Mark Howard; Garrett, Simon Martin

Published in:
International Journal of Human-Computer Studies

DOI:
10.1006/ijhc.1999.0382

Publication date:
2000

Citation for published version (APA):
Lee, M. H., & Garrett, S. M. (2000). Qualitative modelling of unknown interface behaviour. *International Journal of Human-Computer Studies*, *53*(4), 493-515. https://doi.org/10.1006/ijhc.1999.0382

Qualitative modelling of unknown interface behaviour

M. H. LEE AND S. M. GARRETT

Centre for Intelligent Systems, Department of Computer Science, University of Wales, Aberystwyth, Ceredigion SY23 3DB, U.K. email: mhl@aber.ac.uk

When faced with an interface to an unknown system or device humans adopt exploratory interactive behaviour in order to gain information and insight. This paper describes a computer program which probes, observes and models the input}output space of unknown systems. We use a schema concept as the memory structure for recording events and adopt a constructive approach that avoids preprocessing the raw data. We believe qualitative assessments are important in early analysis and employ techniques from qualitative reasoning research in order to capture correlation behaviour. The aim is to gain insight into the nature of an unknown system for guidance in future model selection. A series of experiments and their results are discussed, together with the assumptions and limitations of the method. We suggest further developments for future experiments that appear promising.

1. Exploring structure in unknown devices

Encounters with new and complex man-made systems and devices are a human experience that has become increasingly common in the late 20th century. With the continued advances in technology and the rapidly growing complexity of consumer products, we often have to operate equipment whose functionality we may not fully understand*a well-known example of this is the ubiquitous video cassette recorder. In most cases, when faced with a new device we usually have access to some form of instructions or advice, and also the purpose of the device is known, at least tacitly. Thus, varying degrees of *background* knowledge is available to constrain the scope of possible events and assist our learning of the operation of the system. However, occasionally we meet a system about which we have no prior understanding and we must interact with it in order to gain knowledge of its operation. This sometimes happens with interfaces to unknown software systems as can nowadays be encountered on the internet, common examples being undocumented games or interactive web-based material. Any such unknown system can be called a black-box because we (initially) have no notion of the mechanisms inside the system. This paper reports on an approach to this problem of black-box learning and describes experiments with an exploration and modelling program. Discovering the workings of a black-box is not a new problem and has been previously studied in the domain of electronic and mechanical systems. Our inspiration comes from Weinberg's studies of human behaviour when confronted with physical input/output systems of unknown functionality (Weinberg, 1971). Weinberg constructed a range of increasingly complex electrical and electronic black-boxes and recorded the methods used and extent to which subjects managed to model the boxes correctly. Many of the observations related to human perception and cognitive properties but it was clear that systematic construction of models, starting from the most elementary levels and working upwards, was a bene"cial and probably essential strategy. More recent work has also examined electrical case studies, (Veenman, Elshout & Busato, 1994), and there are many investigations into the psychological and learning aspects of this kind of exploratory behaviour (e.g. Payne, 1991). We do not address human strategies here, nor do we examine any of the human factors involved; instead, we are interested in implementing automatic techniques that can probe the input/output space of an unknown system in order to build up structures that o!er insight into the nature of that system.

We assume a black-box is accessed through some form of interface so that a user is presented with a set of input/output variables that can be changed or observed. In the electrical domain these may consist of switches and meters but in the case of a software black-box we would expect some form of controls and displays. Given such an interface, and a complete lack of information about the purpose, behaviour and operation of the

underlying system, our problem is to automate the exploration and modelling of the structure of the system as seen through the interface.

This problem, sometimes known as system identifcation, has been tackled from several directions. It is important to recognise that various levels of system description have been used in diferent approaches. The most fundamental level is the structural level where states, associations and constraints describe the essential organisational characteristics of a system. Next, the qualitative level can be seen as further description, giving ranges of variables and the overall shape of functional relations. Finally, a fully speci"ed system has a quantitative level where numeric values are available to precisely capture the details of the system. If we accept that a particular choice of structure represents the selection of a model then we can see that decisions at the "rst levels relate to the selection of suitable system models while the later levels are concerned with setting the parameters and "tting trajectories generated by particular models. Most of the literature on classical system identi"cation deals with the "tting of numerical models to data rather than the selection of models. We cannot use these methods for our black-box problem because we have no basis for adopting any particular model structure prior to experience.

A more promising approach to model selection is the work on model-based systems that uses qualitative reasoning techniques. Qualitative reasoning research has developed various methods and notations for handling many forms of system behaviour but there seems to have been little work on the problem of exploring basic system structure and "nding a suitable model. Relevant work includes a system by Richards, Kraan and Kuipers (1992) that abducts qualitative di!erential equations from qualitative system behaviour data and Kay (1997) has obtained results by using multilevel hierarchies, containing structural, qulitative and quantitative knowledge. The use of scale-types to constrain the generation of solutions has been examined by Washio and Motoda (1997). The most relevant example is Haber and Unbehauen (1990), who present a series of examples and case studies comparing methods of structure identi"cation. However, all these system produce models in the form of qualitative di!erential equations suitable for use in the QSIM qualitative simulation approach (Kuipers, 1986). This assumes properties of continuity and di!erentiability in the data, an assumption which we should not accept.

Many modelling systems rely on assumptions of continuity (or local linearity) because they would be mathematically intractable otherwise. In this way, models can be used that approximate di$cult situations to an acceptable degree. But experience of real-world data shows that discontinuities, limited regions of correlation, abrupt breaks in dependencies and other forms of sudden changes in relationships are all quite common. For a fundamental approach to black-box exploration, we must not exclude these features from our modelling scope. For the same reasons, we reject extrapolation, estimation, neural networks and other forms of regression model because our approach is constructive rather than model "tting. That is, we wish to gain insight into the structure of the raw data and not commit to a particular type of model before the input/output relations are fully understood.

We have implemented an exploratory system, called black-box modelling (BBM), which observes the input}output values of an unknown system and attempts to learn patterns of behaviour for prediction and other future use. Our system can be seen as a form of constructive piecewise autoassociator and has similarities with the sparse distributed memory (SDM) arti"cial neural learning system of Kanerva (1988). SDM is a three-layer nearest-neighbour discrete pattern matcher with unlimited storage capacity. The main di!erence with BBM is that it involves no Hebbian weight adjustment scheme. This explains why noise is treated di!erently, as any neural system based on weighted summations will o!er a degree of noise immunity caused by smoothing e!ects.

The next section describes the main features of our approach and the methods adopted. The following section gives details of the modelling system and then a series of experiments are described. After the results have been analysed, the "nal sections discuss the strengths and weaknesses of BBM, summarise our current progress and suggest

future experiments.

2. A qualitative modelling system: BBM

Figure 1 shows the relationship between BBM and an unknown system under test. The box marked OCCR is the mechanism for perturbing the inputs/outputs and is described in Section 3.3. This paper deals with the analysis algorithms; the details of our experiments on control are the subject of a later paper.

2.1. THE SCHEMA STORAGE CONCEPT

The BBM system use a constructive approach to capture the behaviour of the modelled system during continuous interaction. Modelling is thus not a one-shot activity but is continuous and cumulative, starting from an empty structure and building on experience. We use the notion of a &&schema'' as the basic component for storing experience and

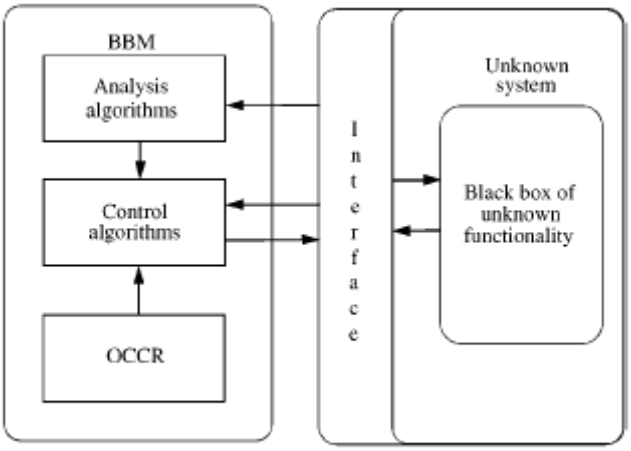


FIGURE 1. The BBM system and an unknown system.

constructing models. Schemas are information structures that an be accessed, combined and coordinated in a number of ways and we de''ne a &&schema'' as follows:

&&A record of a qualitative or quantitative change in the value of one or more output channels, initiated by an action that is activated under zero, one or many *preconditions*.''

Schemas originated in the work of the child development psychologist Piaget (Piaget & Inhelder, 1969). but have been used in computer models as well as in psychological studies. See Arbib (1994, 1995) for an introduction into the literature on the schema concept.

Our use of schemas for recording actions and their contexts is similar to that of Becker (1973), who used such structures for learning sensory-motor relations in robotic applications. The most relevant example of more recent work is the extensive schema experiments in constructive model building of Drescher (1991). We now de''ne schemas more precisely.

For any given unknown system, ;, there must be two or more channels, M;0,2, ;*n*N, at least one of which must be an input and at least one of which must be an output. The set of all input channels is labelled *I* and the set of all output channels is labelled *O*, thus *IXO*'';. The values of the channels in the set *I* at time *t* is I*t* . At time *t*, a particular input channel, *In* , will take one of a set of possible values I(*n*), this set having at least two members. I(*n*)*t* is the value of input *In* at time *t*, where this value is a member of the set I(*n*), and is termed an *action*.

The set of values for all ;'s channels (inputs and outputs), at time *t*, is called the *context*, U*t* , and the terms *precontext* and *postcontext*, U*precont* and U*postcont* , are used to denote the contexts immediately before and after an action. Our schemas record lists of di!erences between the precontext (time *t*!1) and postcontext (time *t*) called the *change-list*.

The precontext and postcontext for a black box, U, with n channels, at time t are:

$$\mathscr{U}_{precont} = \{\mathscr{U}(0)_{t-1}, \mathscr{U}(1)_{t-1}, \mathscr{U}(2)_{t-1}, \ldots, \mathscr{U}(n-1)_{t-1}\} \quad (1)$$

$$\mathscr{U}_{postcont} = \{\mathscr{U}(0)_t, \mathscr{U}(1)_t, \mathscr{U}(2)_t, \ldots, \mathscr{U}(n-1)_t\} \quad (2)$$

The corresponding change-list is

$$\mathscr{U}_{changelist} = \{(\mathscr{U}(0)_{t-1} - \mathscr{U}(0)_t), \ldots, (\mathscr{U}(n-1)_{t-1} - \mathscr{U}(n-1)_t)\} \quad (3)$$

$$= \{\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}\} \quad (4)$$

where $\delta_n = (\mathscr{U}(n)_{t-1} - \mathscr{U}(n)_t)$.

And an action is

$$\mathscr{I}(n)_t \Rightarrow \{\mathscr{U}_{changelist} : t, \mathscr{U}_{precont}\} \quad (5)$$

This can be read, "at time t, an action value of $\mathscr{I}(n)_t$ (on channel I_n) produced a $\mathscr{U}_{changelist}$ of $\{\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}\}$ when precontext = $\{\mathscr{U}(0)_{t-1}, \mathscr{U}(1)_{t-1}, \mathscr{U}(2)_{t-1}, \ldots, \mathscr{U}(n-1)_{t-1}\}$."

For a given schema, *S*, the preconditions, *Sprecond* , consist of two lists of precontexts: the &&for'' list contains precontexts that were observed when it was successful in producing the predicted result; the &&against'' list stores the precontexts that were observed prior to the same action when it was unsuccessful. These provide evidence for predicting the outcomes of an action. Note that &&preconditions'' are tied to a speci"c action as part of a schema, whereas &&precontext'' and &&postcontext'' are merely descriptive of the states of the channel values at any instant.-

2.2. QUALITATIVE MODELLING AND THE MECHANISM OF CORRELATION

Qualitative reasoning is a branch of Arti"cial Intelligence that uses qualitative models in order to explore the theory and application of abstract and approximate representations of physical systems (Faltings & Struss, 1992). Eschewing numeric models in favour of very low granularity qualitative values allows the key features of an application to be highlighted while other details are ignored. Qualitative models often consist of equations or inequalities with values restricted to qualitative symbols, such as [#, 0,!] to represent positive, zero or negative quantities. It has been a remarkable achievement of QR that so much reasoning can be performed on models that represent quantities and their derivatives in this way.

BBM stores both qualitative and quantitative data in schemas. For the output di!erences, the quantitative schemas store the magnitude of the change and therefore depend upon the range of integers used, but for the qualitative di!erences there are only three categories: [#, 0,!], i.e. positive, negative or zero.

If an output channel changes its value at the same time as an input channel, we say the two channels are correlated to some degree. BBM uses the concept of qualitative derivatives to implement this idea. This is best shown graphically, as in Figure 2, which shows a snapshot of two outputs, *A* and *B*, and an input, *C*, plotted against time. Of the channel value changes shown, only a few coincide, i.e. occur over the same time interval; these are between *t*"3 and 4 and between *t*"6 and 7. Channel *A* changes with a change in channel *C* twice, and only changes once independently of channel *C*; channel *B*, on the other hand, only changes with channel *C* once and changes independently of it several

- Taking this approach does have some memory cost, which is needed to store schemas and their statistics, as well as the more important time cost required to search through the lists of schemas in order to update entries. However, Drescher (1991) has shown a similar storage approach can perform e!ectively and e$ciently and we feel our work is also successful in this respect

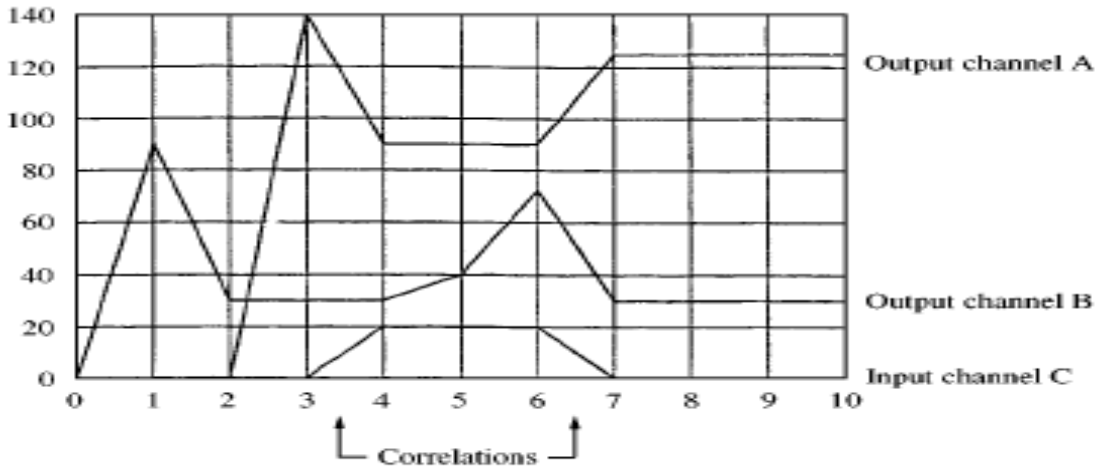


FIGURE 2. A graphical representation of sample channel data.

other times; thus we say there is a higher correlation between channels *A* and *C* than there is between channels *B* and *C*.

As humans, we might test the supposed link between input *C* and output *A* by varying *C* and noting any e!ects on *A* and *B*; this is the type of assessment the BBM system also makes.

The idea of correlation can be expressed using the notation of qualitative derivatives. The qualitative "rst derivative is given by: *qdir*( )"[d/d*t*], i.e. the sign of the change in direction of a variable. We can then express the correlation in Figure 2, thus: *qdir*(*A*)"!*qdir*(*C*) between *t*"3 and 7. Note that higher order derivatives are not involved at this stage.

2.3. EXPLORATION BY EXTERNAL EXCITATION

Experiential data are accumulated during the operation of BBM and the schemas formed are used to guide future explorations.

In order to maximise the information gained about the unknown system being analysed, it is important that the BBM system should use its control over the inputs to generate the schema input information in a way that aims to cover as much of the input state space as possible. Many theoretical states are not possible in reality due to the functionality of the unknown system. Despite the inevitably sparse covering of the state space, BBM should attempt to cover the input state space as evenly as possible by repeating the same actions in di!ering circumstances.

We used two methods to explore the input state space: (1) ramping of inputs incrementally through a range in sequence, and (2) detect range limits by exponential probing combined with normally distributed random values within the discovered ranges. In the "rst method the input values are ramped from zero up to set limit (typically 20) and then down to the inverse of the limit. If the range of the input's values were greater than the set limit, the system "nds them by attempting to set each input to an ever increasing power of two up to a maximum value (typically 32 678). Once the inputs' minimum and maximum limits have been found, the BBM system continues its analysis of the unknown system by choosing random values between the minimum and maximum limits. The number of random values chosen can either be a "xed number or a percentage of the range between the minimum and maximum values of the input.

2.4. ASSUMPTIONS AND LIMITATIONS

No exploratory system can avoid embodying assumptions which impose basic limitations on its use and performance. For these "rst experiments using schemas and techniques from qualitative reasoning, we recognize that some primitive constraints have been accepted. The intention is to lift these in later experiments after bene"ting from the experience and insight gained from the results. In particular, the main assumptions and limitations, are as follows.

- All channel values are discrete values. These are usually speci"ed as an integer range, with binary variables being a special case.
- The version of BBM reported here allows only one input to be active at a time; a new version has successfully relaxed this constraint (Garrett & Lee, 1999).
- Sequential actions and delayed response models cannot be constructed. This is because the system described only accesses the immediate precontext.
- In some systems, certain channels are able to behave as both inputs and outputs according to the direction of the applied actions. However, this version of BBM assumes that no channels are bidirectional.

In later experiments we have addressed all these issues which are brie#y discussed in Section 7.

3. The three stages of BBM modelling

There are three main stages in the present system. Stage 1 discovers the input/output directional character of each channel of the system being analysed; Stage 2 is a learning phase where schema are created to record the e!ects of the inputs on the outputs; and Stage 3 recalls relevant schemas for a given context in order to access their validity. Schema learning continues throughout Stages 2 and 3.

3.1. STAGE 1: INPUT/OUTPUT ANALYSIS

This stage determines which channels can or cannot be *directly* a!ected. In order to discover the nature of each channel, BBM applies a positive and negative change to the value of each channel in turn. Given that no more than one input is non-zero at any time *t*, a channel's response to an attempt to change each channel in turn will reveal it as being:

An obvious input: if a channel's value is directly alterable by the BBM system, and such a change in value a!ects other channels, it must be an input channel.

A deduced input: if a channel is directly alterable by the BBM system it cannot be an output since, by de"nition, outputs are controlled only through the functionality of the unknown system. Therefore, it must be an input even if there are no observed e!ects on other channels.

A disabled input/rarely responding output: if a channel is not alterable directly it may still be an input that requires a set of circumstances before it will become alterable. Alternatively, it may be an output channel that required special precontextual circumstances before it becomes changeable by an input channel. We consider such channels to be outputs.?

Disconnected: channels which do not respond at all because they are not connected will appear to be disabled inputs or rarely responding output channels and will be treated as such. Assuming they are outputs will not a!ect the analysis.

An obvious output: if a channel changes value at the same time as another channel is being directly altered by the BBM system, it will be assumed to be an output channel.

3.3.1. An illustration of stage 1 analysis

Consider an unknown system with "ve channels. The BBM system will interact with the unknown system as shown in Table 1 in order to discover the directional nature of each channel. Taking the initial channel values as zero, the BBM system tries to change each channel in turn, "rst positive and then negative, relative to the initial value. The BBM system sends this requested change to the unknown system as a set of *n* values, where *n* is the number of channels of the black box; in this case, "ve.

This set of requests is shown as *Req<al0* } *Req<al4*, in Table 1. The resulting channel values, also shown, are the postcontext for the set of *Reg<als* in the same time interval; the precontext is the set of channel values for the previous time interval. For example at *t*"2 the precontext"M1, 0, 0, 0, 0N, &&action""[*Ch*0P!1], and postcontext"
M0, 0, 0, 0,0N.

-Ideally, these channels would be considered inputs until there were any evidence that they are outputs (i.e. they are changed by another channel). In fact, we assume they are outputs for two reasons. Firstly, disabled inputs are rare and ignoring them will only limit the applicability of BBM very slightly. Secondly, assuming a channel is an input until proven otherwise demands either that all outputs should show their directional nature during BBM Stage 1, which is unrealistic since many outputs require a complex set of conditions before they will change; or that BBM Stage 1 continues throughout all three BBM stages, which adds unnecessary complexity to the rest of the analysis stages

TABLE 1
Sample data for a five-channel system

Time	0	1	2	3	4	5	6	7	8	9	10
ReqVal 0:	0	1	−1	0	0	0	0	0	0	0	0
ReqVal 1:	0	0	0	1	−1	0	0	0	0	0	0
ReqVal 2:	0	0	0	0	0	1	−1	0	0	0	0
ReqVal 3:	0	0	0	0	0	0	0	1	−1	0	0
ReqVal 4:	0	0	0	0	0	0	0	0	0	1	−1
Channel 0:	0	1	0	0	0	0	0	0	0	0	0
Channel 1:	0	0	0	1	0	0	0	0	0	0	0
Channel 2:	0	0	0	0	0	1	0	0	0	0	0
Channel 3:	0	0	0	0	0	0	0	0	0	0	0
Channel 4:	0	0	0	0	0	1	0	0	0	0	0

From this we can see that the "rst three channels respond directly when attempts are made to change their values to 1; the fact that they do not respond when an attempt is made to change them to!1 is irrelevant since any channel that shows *any* direct change cannot be an output: these channels are inputs. Channel 3 does not change at all and the BBM system assumes it is an output. The remaining channel does not appear to be directly controllable, meaning we will also assume it is an output, and this is con"rmed by its change in value at *t*"5, when channel 2 was set to 1. In the terminology used above, channel 2 is an &&obvious input", channels 0 and 1 are &&deduced inputs", channel

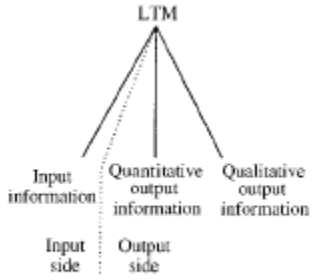


FIGURE 3. The empty LTM structure.

3 is a &&rarely responding output" or a &&disabled input" and channel 4 is an &&obvious output".

3.2. STAGE 2: THE GENERATION OF SCHEMA INFORMATION

Schemas are stored in a structure that we call long-term memory (LTM). This is a tree-like, hierarchical data structure that begins as an empty structure as shown in Figure 3.

To aid comparison with other schemas, and to facilitate retrieval, schemas are stored in LTM in two related structures: *schema input information* and *schema output information*. We now examine these schema components.

Schema input information is stored hierarchically on the input side of LTM. It is "rst sorted by input channel number; then sorted by value; and under each value details are stored about the preconext, list of changes and time of creation. This is shown in Figure 4. The brackets show how the context and e!ects of an action are separated from the action itself.

Achieving control over a black-box means being able to control the output channels. This implies being able to accept and achieve a task such as, &&change channel *On* by *d*".

To esciently fufil such requests the system generates a heirarchically structured list of output channels and associated changes. Under each change value the BBM system stores indexes to the actions that caused the change so that the information about those actions may be quickly retrieved. This is shown in Figure 5. This structure is used for

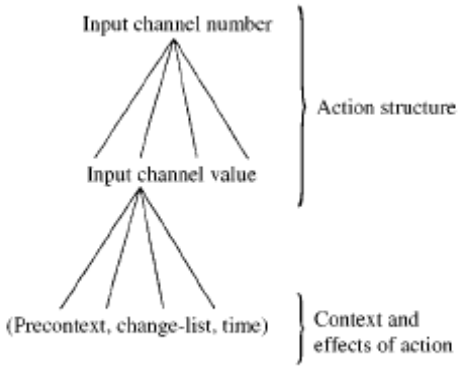


FIGURE 4. The structure of the schema input information.

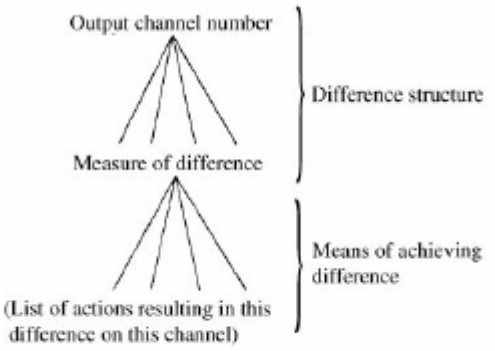


FIGURE 5. The structure of schema output information (qualitative and quantitative).

both qualitative and quantitative schema output information.

After some time, schemas have been formed for each action initiated and indexed into the LTM, whose complete structure is shown in Figure 6. Some typical data from our implementation of the schema input construction process are shown in Figure 7 where the structure described above can be discerned. This is an excerpt of the schema input information for Input Channel 0, showing most of the information stored under value 0. The lines under &&Value"0" consist of the precontext, the change-list and the time. For example, the "rst line records that the postcontext was M1, 0, 30, 3N, the change-list was M0, 0,!5, 0N and the time *t* was 45. In this case *I*"MU0N#MU1N and *O*"MU2N#MU3N, thus the new input values, M1, 0N, will initiate a further change list for the next time step. There is also a line for each new input, beginning &&Input Channel...", that lists all the values this input has been seen to take; in this case every integer value from 0 to 19.

A similarly display of some schema output information is given in Figure 8. These diagrams are complicated by the method of implementation which separates schema elements. Also, the process of extracting change data from the input stream at the interface is not easily discerned from the implementation data structures.

Schemas become more accurate the more they are used since each time an action is initiated, new data are converted into schema information. Exactly how this helps the BBM system to choose the best schema for a particular context is discussed next.

3.3. STAGE 3: SCHEMA RETRIEVAL AND SELECTION

&&Retrieval" refers to the gathering of potentially useful schemas fromLTM via the output side indices, whereas &&selection" refers to their subsequent ordering by some "tness

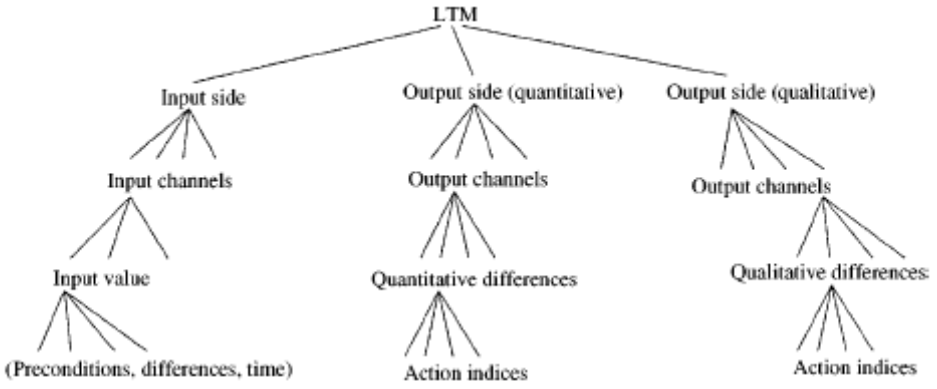


FIGURE 6. The filled structure of LTM.

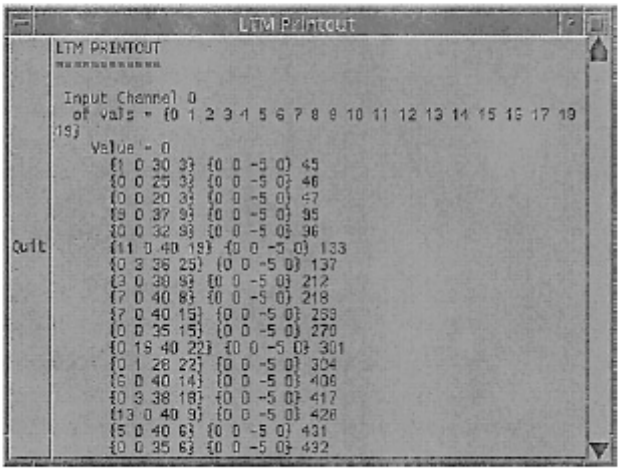


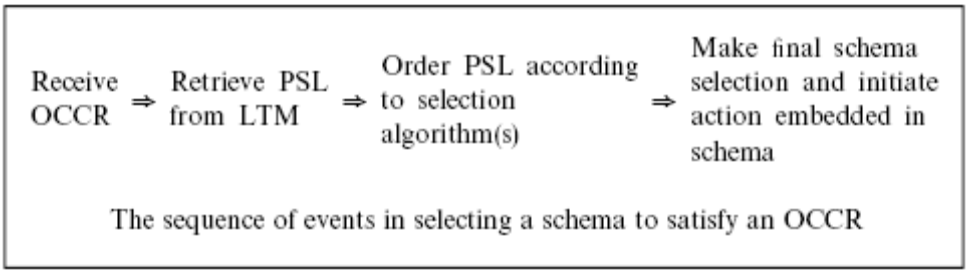
FIGURE 7. A display of part of LTM showing schema input information.

function and then choosing the most suitable schema. We stress that the BBM system attempts to "nd schemas which will match the *current* context rather than changing the current context to match a reliable schema [as Drescher'sA (1991) work did].

BBM aims to select a schema from LTM that contains an action which changes the inputs so that it will produce a desired change in an output value. This desired output change is called the *output channel change request* (OCCR). An OCCR is only allowed to request a change in the value of one output channel at a time, although such a change will often cause side-e!ects on other output channels. If a goal requires several output channels to be changed then each output channel must be changed by an individual OCCR, although the BBM system does check whether the side-e!ects of changes on one output channel successfully achieve OCCRs on other output channels. During this stage, OCCRs are created by random generation.

At retrieval the schemas are reconstituted from their component parts in LTM, i.e. the preconditions (the &&for" and &&against" lists, as described in Section 2.1), the action that formed them and the output information index matching the OCCR. The list of all actions that have resulted in the desired change in the past is called the *possible schema list* (PSL). Next, the PSL is ordered in various ways using one or more of the selection algorithms described below. The resulting list or lists are resorted, and the best schema is selected and its action is initiated.

In diagrammatic form:



The sequence of events in selecting a schema to satisfy an OCCR

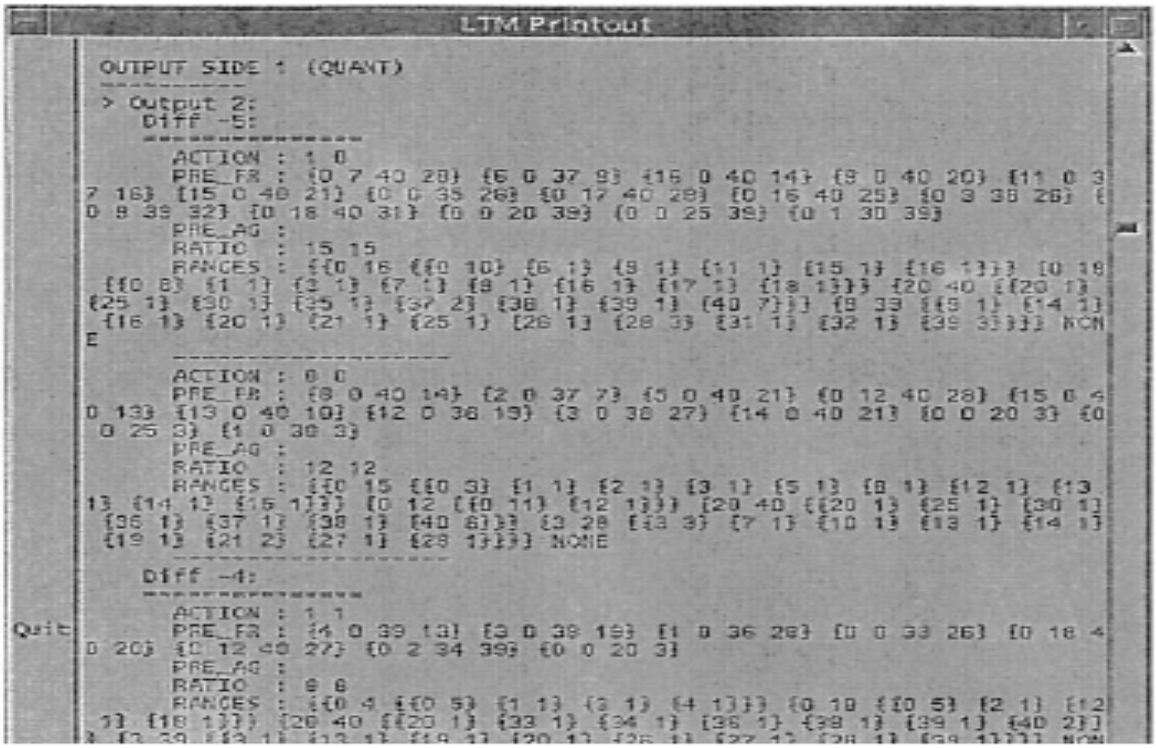


FIGURE 8. A display of part of LTM showing schema output information.

The sequence of events in selecting a schema to satisfy an OCCR

There are three main selection algorithms, *for-against ratio maximising*, *hyperspace distance measuring and range-value matching*, as well as two random algorithms.

3.3.1. Test algorithms

Two random algorithms, `alg-12` and `alg-13`, were used as experimental controls. These simply select a new action for each new OCCR randomly, either from the existing schemas or from the PSL, respectively. The results give a baseline for the learning algorithms.

3.3.2. The for}against ratio maximizing (FARM) algorithm

The FARM algorithm, `alg-21`, simply counts the number of &&for'' and &&against'' precontexts in the preconditions structure for each schema, and creates a ratio of the number of &&for'' precontexts divided by number of &&against'' precontexts. (If there are no &&against'' precontexts the value of 0.1 is used rather than zero.) This gives a list of each schema's chance of success, based on previous experience. The higher the ratio, the more likely it is that the schema should achieve its predicted result. The BBM system then chooses the action that has the best apparent chance of achieving the goal change. The FARM selection algorithm is coarse but simple and fast.

3.3.3 Hyperspace distance (HD) algorithms

The hyperspace distance (HD) algorithms, `alg-31` and `alg-32`, treat the precontext elements in the precondition's &&for'' and &&against'' lists as points in hyperspace. The HD algorithms assume that being close (in Euclidean distance) to the precontext of a previously successfully achieved OCCR will make it more likely that the OCCR will be achieved again; and, conversely, being close to the precontext of an unsuccessfully achieved OCCR, less likely. This is e!ectively a *k*-nearest-neighbour approach with *k*"1.

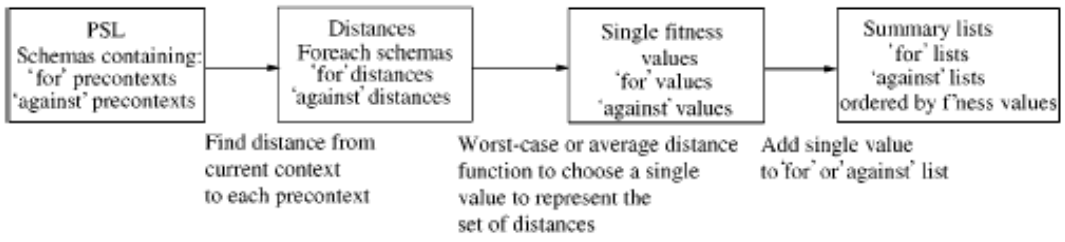


FIGURE 9. Creating the "for" and "against" precontexts lists for the hyperspace distance algorithms.

The HD algorithm calculates the Euclidean distance over all dimensions from the current context to each precontext element in the precondition structure of each schema in the PSL. The &&for'' and &&against'' lists are processed separately.
In algorithm `alg-31`, the *worst-case* distance from each schema's &&for'' and &&against'' list elements is calculated. In the case of the &&for'' list this is the furthest distance from the current context that has successfully resulted in the OCCR; in the case of the &&against'' lists it is the closest distance that has led to failure to achieve the OCCR. This is a pessimistic but conservative strategy.
In algorithm `alg-32`, the BBM system takes the *average* distance from all the &&for'' and &&against'' precontext elements. This is more optimistic and aims to promote schemas whose preconditional elements are generally favourable but which would be overlooked if they contained even a single bad precontextual element.
Both these approaches are summarized in Figure 9.

3.3.4. Range-value matching (RVM) algorithms
More reliable predictions of the e!ects of a given schema should be available if the BBM system considered the individual elements of a precontext, rather than the precontext as a whole. The range-value algorithms use the *range* of values, *Rx*, for a particular channel of the precontext and the *frequency* of the values within that range.
We could use these data simply to promote schemas whose &&for'' precontext ranges encompass the current context's value for each of the *m* output channels, and whose &&against'' ranges do not. For the &&for'' list, if a channel value is &&in range'' the BBM system assumes it is more likely that the schema with this range will obtain the desired OCCR, and conversely for &&against'' ranges. With this approach, the BBM system would consider each channel separately unlike the hyperspace distance algorithms.
However, algorithm `alg-41` attempts to narrow the scope of this and all the previous algorithms. We need to identify channel values that are essential to a schema's reliability. Output channels that have narrow ranges in the &&for'' precontextual data express a high need to match to that range, but ranges that are very wide are not so important.
Moreover, if a current context's output channel value matches a range value exactly, as well as being in range, this should be registered as an even better match; especially if that value has been seen many times compared to other values in the range. Thus, two measures are to be maximised when using `alg-41`: (a)=*r* , which describes how well the range encompasses the current context channel value. This will be the width of the range if the point falls within the range; otherwise it is given by the range multiplied by the distance from the nearest end of the range. This is expressed by the function *dist*(*r*) for a distance *r* from the nearest end of a range *R*, and (b)=*v*, which expresses how well the current value for a channel matches any previously seen values, calculated by dividing the frequency, *fx*, of that value by the sum frequency of all values, +*n*~1

i/0
fi .

Overall, we have

$$\mathscr{W} = W_r + W_o \quad \text{or} \quad \mathscr{W} = dist(r) + \frac{f_x}{\sum_{i=0}^{n-1} f_i}$$

Once the &&for'' and &&against'' ordered lists are formed by one of the algorithms above, the selection of the winning schema, whose action will be initiated, can be made. The BBM system simply selects the schema with the highest value returned from `alg-12` to `alg-41`. (Selecting randomly from a small number of top schemas was found to be less reliable than choosing the single best schema.)

4. The experiments

The experiments and results described here are all taken from Garrett (1997). This thesis contains the background for a constructive schema-based approach, details of a range of experimental black-box systems, more on the above set of experimental algorithms and information about implementation methods. Also included are the full set of collated results and their analysis and appraisal. Full details on all these aspects can be found in Garrett (1997) but we present the main results and signi"cant "ndings here.

The BBL system was implemented in the high-level scripting language Tcl/Tk within a Unix environment. The experimenters interface is shown in Figure 10.

We performed a wide range of experiments using all the above algorithms on a series of target systems. Following normal experimental practice, the target systems start with almost trivially simple designs and increase in complexity until real applications are reached. The simple cases are valuable because they are small enough that their outputs can be calculated; thus, the implemented algorithms can be thoroughly tested and validated by detailed examination and checking.

All the experiments were grouped according to whether they had: binary- or integervalued variables; single, double or multiple numbers of input and output channels; and static or dynamic internal functions. Static systems have combinatorial functions while dynamic systems incorporate output values from the previous time step into their function. This gave 72 groups: there are six possibilities for input (or output) con"guration and so there are 36 cases for each of static and dynamic systems. Our experiments were designed to cover a representative set of these cases.

The experiments fall into three types.

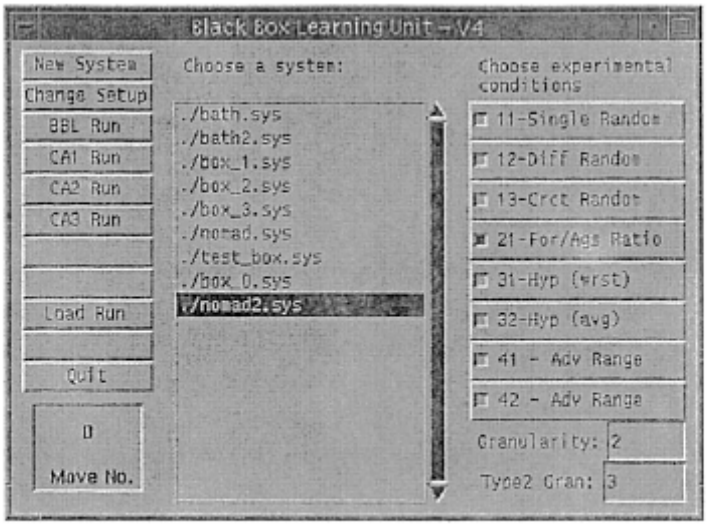


FIGURE 10. The experimental controls for BBL.

Simple systems. These include primitive systems (e.g. with a single output simply related to a single input) and are useful for testing and veri"cation of the operation of BBM. The more interesting systems used input/output functions with linear relations, summation e!ects and thresholds.

[1]*est cases*. A series of cases were designed to examine the performance of BBM on noisy, meaningless and low-grade data. These are

called &&testboxes" and we show results
for three examples: a system that can not be learned (purely random output); an example with added noise; and an intermittent error case.
Simulated real systems. Simulations were constructed to provide black-boxes with behaviours approaching that found in real-world systems. We experimented with a bath "lling example, a car control system, a tra$c problem and a mobile robot application. We give details of the bath problem here.
As our main example of a realistic dynamic system, we used a bathtub that can be "lled with water by taps at various temperature. We investigated three variations of the bath-"lling experiment: a two-tap version with depth and temperature sensing, `bath1`; a three-tap version, (cold, warm and hot), `bath2`; and a two-tap version with sensing of the frequency of operation of the taps, `bath3`. In all versions, the bath is 40 units deep maximum and the water is pumped out at a constant rate of 5 units per time interval (regardless of water depth). Instant mixing of di!erent temperatures is assumed. There is a depth gauge and a temperature reading device. The taps (inputs) work independently of one another and all have maximum and minimum #ow rates of 19 and 0 units of depth per time interval. The &&cold", &&hot" and &&warm" taps control the entry of water at 3, 40 and 203C. In the tap frequency version, the relative frequency of the last two episodes of cold- and hot-tap activation is recorded. For example, if the cold tap has been active for the last 4 time units, and immediately previously the hot tap had been active for 6 time units, the interleaving frequency is 4/6 ("2/3). The next time the hot tap is activated its frequency count is reset to zero, and so on for both cold and hot taps. The task is to keep the water at a given level and temperature (and, if applicable, at a given interleaving frequency): a di$cult enough task to be challenging and non-trivial.

5. Analysis of results

Each experimental unknown system was processed by each of the six PSL selection algorithms and a set of 50 qualitative attempts and 50 quantitative attempts constituted one run. A full set of results for a box consisted of 10 runs, the results of which were then

averaged. Thus, each system was subjected to 500 qualitative and 500 numeric change requests.
As the generation of change requests involves a random element, the experiments were calibrated by repeating over full set of results; the average deviation between sets of results was 2.08% with only 5 out of 91 di!erences greater than 5%, all of which were less than 9%.
The points visited in the state space were plotted from the results to ascertain the coverage produced during exploration. In all the simple boxes, the density was acceptable but in most of the real applications the coverage was very sparse or patchy.
For each experimental system, we computed a table of success metrics, giving the level of qualitative and quantitative success as the percentage of trials producing correct results. All results for each system were plotted as graphs of algorithm version against percentage success; with separate plots for qualitative, quantitative and average level of success. The full set of results are tabulated and displayed in Garrett (1997). For space reasons, we show here the more complex (and interesting) cases using the non-test algorithms, `alg-21 to alg-41`.
The success rates of the more exacting algorithms increase somewhat with the number of input channels. This makes sense since the BBM system does not have to rely on the schemas formed around the actions of one channel. Because the OCCRs are generated automatically, the results include cases of impossible actions, i.e. requests that are out of range of the current variables. Thus, the results of trials should not be expected to be 100% correct; a reasonable "gure is over 90%.
In the single-channel binary-valued systems, the problem space is so small that complete mapping of input}output behaviour into schemas is achieved quite rapidly. These were used to validate the implementation and the experimental design.

The three testbox experiments are all based on noisy variants of a previous test case. In testbox-random, we investigated how BBM copes with a system that contains two outputs that are not controllable (the outputs are set to random values). In a similar way, the testbox-noisy showed he e!ects of adding noise to an input's value (random noise at 50% of max signal level) and in testbox-loose a &&loose output connection" was simulated by returning zero randomly for one-third of the output readings.
The results, shown in Table 2, all show relatively poor performance as could be expected. testbox-random shows the lack of any structure being detected by BBM. With a three-valued qualitative representation random data has a 33% chance of success*this is clearly seen in Table 2, while the quantitative results are showing almost zero matching. With a fairly high noise level testbox-noisy performs only slightly better on the quantitative results while performing remarkably well on qualitative data. Testbox-loose, with an intermittent disconnection, proves more di$cult for qualitative matching but produces better quantitative results. This is as expected as the data is noise-free when not subject to the fault; thus the repeatability is higher.
The bath results cover part of the experiments concerned with multiway relationships and non-binary channels. The bath-1 results are shown in Table 3. The separate success rates of the two output channels indicate that, with numeric matching, the BBM system "nds it much easier to model channel 2 (water depth) than channel 3 (temperature). There is little to choose between the algorithms but channel 3 fails on numeric matching.

TABLE 2
Results for the testboxes: random, noisy and loose

SYSTEM: testboxes		random		noisy	loose
		Channels		Channels	Channels
Algorithm	Result type	1	2	1	1
Alg-21	Qualitative	31.5	32.0	72.426	67.537
	Numeric	1.552	2.561	5.246	28.001
Alg-31	Qualitative	36.0	35.0	74.35	59.472
	Numeric	1.021	2.094	9.522	27.418
Alg-32	Qualitative	30.0	26.5	68.367	59.501
	Numeric	1.042	1.064	7.301	23.571
Alg-41	Qualitative	34.5	33.5	77.469	65.302
	Numeric	1.01	1.032	8.188	31.937

TABLE 3
Results for the bath-1 and bath-2 systems

SYSTEM		bath1			bath2		
		Channels			Channels		
Algorithm	Result type	2	3	4	3	4	5
Alg-21	Qualitative	98.421	99.4	—	99.706	99.4	—
	Numeric	93.584	12.412	—	93.519	15.665	—
Alg-31	Qualitative	96.716	90.717	—	98.492	78.0	—
	Numeric	91.715	11.668	—	94.289	14.416	—
Alg-32	Qualitative	96.806	98.179	—	99.394	87.763	—
	Numeric	94.964	13.204	—	94.102	15.074	—
Alg-41	Qualitative	97.029	98.796	—	94.465	97.0	—
	Numeric	91.932	12.304	–	95.833	16.629	—

The results for the `bath-2` and `bath-3` systems are very similar although the BBM system has more trouble modelling the `bath-3` systems with its unusual interleaving concept between inputs (see Table 4).
The results for the remaining applications were all very similar and not very high*the qualitative results were the best, with success rates of 78% compared with around 50% for the random algorithms. We report on the mobile robot experiments which were typical for all these cases.
The robot used was a Nomad commercial mobile robot that could move either forward at variable speed or rotate about its base. We had a simulator available for this robot system and we used this to form the black-box interface. The actuation and sensing variables selected as interface channels consisted of two motor channels for forward and rotate, 16 sensory channels for sonar object detection and three sensory channels for orientation and position-sensing.

TABLE 4
Results for the `bath-3` system

SYSTEM		bath3				
		Channels				Mean
Algorithm	Result type	2	3	4	5	
Alg-21	Qualitative	99.7436	98.6	83.4	85.6	91.8359
	Numeric	81.3942	14.3681	11.2	11.0	29.4905
Alg-31	Qualitative	94.0485	97.9795	83.6	66.4	85.507
	Numeric	96.3923	8.87806	12.4	11.1999	32.2176
Alg-32	Qualitative	92.927	99.3792	63.4	68.6	81.0766
	Numeric	94.3534	10.4919	12.4	9.8	31.7614
Alg-41	Qualitative	93.9148	99.3918	77.8	80.0	87.7766
	Numeric	90.1814	10.1896	12.0	10.2	30.6427

The results can be plotted in a chart that shows the usage of each variable value and the density of the plots indicate the coverage of the state space achieved. In the robot experiments, this showed very localised exploration with patches of activity in a generally sparse space. There are two reasons for this: "rst the OCCRs may generate values that are impossible for the application and secondly the probing mechanism has no sense of direction and does not learn from the structure already discovered. We believe this accounts for the rather inconclusive results.

6. Discussion

The aim of this work was to examine methods for probing an unknown interface in order to explore its behaviour in terms of dimensionality, continuity and other structural aspects. Data collected from unknown systems of any complexity may be sparse, incomplete and contain many discontinuities and disruptive aspects. This "rst set of experiments has entailed some fairly severe constraints and we summarise these and their e!ects below:
No prior modelling bias. We have avoided the temptation to "t incoming data to a preselected model; thus we do not use curve "tting, neural networks or any other forms of non-linear regression technique.
No explicit generalization. We have adopted a memory-based approach in which all data points are stored if they have not already been experienced and captured. This does not employ any weighting or smoothing factors and hence there is no explicit treatment of noise.
No explicit interpolation or extrapolation mechanisms. We have avoided the continuity assumption that correlations exist in the regions between data points. This assumes the data are not continuous but has regions of irregularity and changing relations between the inputs and outputs. It can also be sparse and impoverished.

In the work described, most of the quantitative results prove ine!ective at detecting structure in the data. This is because of our assumptions cause the current BBM system to treat individual channel values as distinct entities. This invokes the &&curse of dimensionality '' as the input space becomes so enormous and so sparse. Nevertheless, this characterizes the nature of the data we aim to explore and explains why we approached the problem with a memory-based technique and nearest-neighbour selection. The qualitative results produced higher levels of success because mapping an extremely sparse space is much easier with qualitative variables.

E!ectively, noise does not exist if it is below the resolution of the descretized variables. In the case of qualitative changes, i.e. [#, 0,!], quite high levels of noise can be tolerated provided an adjacent qualitative region is not entered. This explains why the results for noisy signals are so much better than might be expected.

The notable features of BBM are as follows:

Qualitative structure. Despite the above restrictions, qualitative variables have proved very e!ective in capturing general but gross structure. Thus qualitatively similar data, stored in the form of schemas, are reused in similar situations wherever appropriate. A key feature that contributes to the reuse of schemas is the matching of di!erences, rather than target values.

No prior bias. Notwithstanding the limitations above, the lack of model bias means that, in high scoring systems, the schemas constructed will authentically re#ect the structure of the data. Hence, a collation of the qualitative schemas will provide insight into the nature of the data and give guidance for the selection of a more quantitative model.

Reproducibility. Relating to the above point, as there is no statistical estimation involved, all learned data patterns and regularities can be regenerated from schemas exactly.

Constructive piecewise approach. Our system can be seen as a piecewise autoassociator. Such systems attempt to model the unknown black-box function f by constructing a collection of functions: h_1, h_2,…, h_m each of which is only partially de"ned on a subset of the input space. In many piecewise function decomposition systems the constructive functions are linear; however, although we could easily have used this to add linear interpolation to BBM, we decided that it should be included only in later versions.

Incremental learning. Unlike the need for o!-line training in SDM, there is no separate training phase in BBM; schemas are continuously constructed. New schemas are learned whenever the system has insu$cient experience to cover the current data.

Continuous validation. As the data are not modi"ed by storage there is no need to validate the state of the system after a series of learning runs. Validation automatically occurs whenever a new set of data recalls an appropriate schema.

Separation of storage scheme from processing method. The LTM data structure proved valuable for supporting experimentation with di!erent schema selection algorithms. This can readily facilitate new selection methods or enhancements such as linear interpolation.

7. Addressing the limitations of BBM

We have noted the limitations of our current system and have carried out further investigations to "nd ways in which these may be overcome. One severe limitation is that only one input is allowed to be non-zero at any one time interval. Multiple simultaneous actions produce the problem of deciding whether the changes in output values are a result of the coincident, *individual* e!ects of the inputs or a function of the *joint* inputs. One way of investigating this problem is to treat combined inputs as a single *complex input* and to compare the e!ects of this complex input with the e!ects of each of its component parts. In order to deal with this problem, we performed some experiments with an interface through which the inputs were accessed via incremental bu!ers. Extra components in the interface would increment the internal inputs if an input was positive, and decrement it if negative. This allowed several channels to be non-zero at the same time, provided that their values do not change simultaneously. We achieved control of the bathtub systems with this arrangement and will report on results later.

Another limitation of the current BBM system is that it cannot model channels that are bidirectional in character. However, this can be achieved by characterizing the input channels as before but with the provision that *any* channel might possibly be an output even if some or all are also inputs. The structure of LTM will then need to store details about an individual channel's input and output behaviour in both the input *and* output side of LTM, but otherwise remains much the same. One of the problems of this approach is that it assumes all channels could behave as outputs. Thus, there will be more output information gathered and output analysis will be slower.
BBMcan be extended so that *sequences* of actions, that lead to desired events, can then be discovered. Using the recorded time index of each context element it is possible to compare more than just the immediate precontext before an event. Given that BBM records the entire history from the beginning of its analysis of an unknown system it can detect common strings of actions of arbitrary length. However, using sequences of actions and preconditions requires devising a way of "nding commonalities in strings of preconditions for sequences of a given size in order to capture the goal of the complete sequence. This can be done for small sequences but otherwise raises complex pattern matching problems

8. Future work

There are two main areas in which we are continuing work on BBM. First we need to develop methods for presenting the LTM repository in a way that is helpful for perceiving the structures that have been collected. This mainly involves processing and collating the LTM contents suitable for presentation. It will be important to "nd the most appropriate viewpoint to bring out the captured relationships and modern visualization techniques may be useful here. Secondly, as was hinted at by Figure 1, we have extended BBMfor use as a controller to maintain a given goal state in an application. Experiments are under way to show how several precontexts can be pursued as a control action.
There are also a number of further experiments that appear promising. Hierarchical schema structures, called &&composites", have been used by Drescher (1991), and Becker (1973) also proposed a similar meta-schema device. It seems clear that such higher order schemas may o!er methods for compression of base schemas into more general data abstractions. Other issues include the above-mentioned use of piecewise interpolation for quantitative schema construction, the exploitation of second-order di!erentials of input/output data and further treatment of compound actions.

Regarding real-world target applications, as mentioned previously, a possible application is an &&unknown" software system such as an interactive service or game on the world wide web. This would provide another series of experiments but we have not yet connectedBBMto such a &&black-box". Our experience with real data in the mobile robot experiment was not very conclusive. The main di$culty was that a great many trials are required to collect enough data to e!ectively populate the problem space. Our design of BBM was intended to navigate through the state space but the density plots showed that the exploration strategy appeared ine$cient. It seems that our method of range probing (Section 2.3) is insu$cient for driving the navigation process through complex state spaces for certain domains. This requires a more intelligent search or guidance process that needs further investigation.

9. Conclusions

Our experiments have explored the role of qualitative representations in a piecemeal constructive approach to mapping unknown systems without predetermined model structures. We have emphasized the use of raw data, involving discontinuities, sparsity and fragmentation. These decisions have imposed severe constraints and BBM has several shortcomings. However, the results illustrate the potential of our approach and suggest further developments that may prove valuable in automatic modelling experiments.

The &&curse of dimensionality'' has not been addressed directly; that was not our intention, instead by recording patterns in LTM we gain insight into the behaviour of the unknown system that will aid future modelling. In that respect, our LTM structure has similarities with the SDM learning model of Kanerva (1988) and other distributed memory models.

We notice how qualitative values have introduced some limited noise immunity and believe higher-order qualitative di!erentials will add enhanced representation of the key model features. We did not address noise and precision issues explicitly but regarded any noise levels as being lower than the resolution of the discrete variables.

We have seen the role of qualitative representations in the trade-o! between speci''city and generality. Despite their imprecision, qualitative models have proved valuable in many application areas for abstracting or capturing the essence of system behaviour.

The three main algorithms used in the experiments selected schemas on the basis of: the ratio of numbers of positive to negative examples for the change desired; the nearestneighbour
(Euclidean) distance; and an interval measure of matching on each channel.

The ''rst two algorithms can only o!er a crude approximation of a schema's ''tness because they treat the schema precontexts as multi-channel entities. The range-value matching algorithms attempt to match each element of the context to the individual channel elements of the precontextual information. However the nearest-neighbour strategy is very suitable for this problem space and we suggest the application of the hyperspace distance algorithms to individual channels in combination with the range matching method will give further progress.

Unlike other approaches, our method does not assume any prior structure but o!ers a way of automating the exploration of unknown systems. This could be very valuable as a ''rst stage process that gathers data, determines structure and suggests models for the system under examination. The longer term bene''ts are in building tools that perform this function and will have considerable use within agents that explore internet and web-based services. Our contribution is to illustrate the role of qualitative representations and promote a constructivist approach. We hope this kind of work will continue and will shed further light on the qualitative structure of problems in a way that relates to the &&intution and insight'' that so characterizes human thinking, especially when exploring unknown systems.

References

ARBIB,M. A. (1994). Schema theory: cooperative computation for brain theory and distributed AI. In V. HONAVAR & L. UHR, Eds. *Arti*,*cial Intelligence and Neural Networks*: *Steps toward Principled Integration*, pp. 51}74. Boston: Academic Press.

ARBIB,M. A. (1995). Schema Theory: from Kant to McCulloch and beyond. In R. MORENO-DIAZ & J. MIRA-MIRA, Eds. *Brain Processes*, ¹*heories and Models*, *An International Conference in Honor of* =. *S*. *McCulloch 25 years after his death*, pp. 11}23. Cambridge, MA: The MIT Press,

BECKER, J. (1973). A model for the encoding of sensory-motor schemas in a mobile robot. In R. SCHANK&K. COLBY, Eds. *ComputerModels of* ¹*hought and* ¸*anguage*. San Francisco: W. H. Freeman.

DRESCHER, G. L. (1991). *Made-up Minds*: *A Constructivist Approach to Arti*,*cial Intelligence*. Cambridge MA: MIT Press.

FALTINGS, B. & STRUSS, P., Eds. (1992). *Recent Advances in Qualitative Physics* Cambridge, MA: MIT Press.

GARRETT, S. M. (1997). *A schema-based*, *unsupervised learning approach to controlling a system of unknown functionality*. Ph.D. Thesis, University of Wales, Aberystwyth.

GARRETT, S. M. & LEE M. H. (1999). A case-based approach to black-box control learning, In *Proceedings of the International Conference on Computational Intelligence for Modelling*, *Control and Automation*, (*CIMCA* 1*99*), pp. 372}377. Vienna: IOS Press.

HABER, R. & UNBEHAUEN, H. (1990). Structure identi''cation of nonlinear dynamic systems*a survey on input/output approaches. *Automatica*, 26, 651}677.

KANERVA, P. (1988). *Sparse Distributed Memory*. Cambridge, MA. USA: MIT Press.

KAY, H. (1997). Robust identi''cation using semiquantitative methods. In *Proceedings of the*

Safeprocess 1*97*, *IFAC*, pp. 288}293.
KUIPERS, B. J. (1986). Qualitative simulation. *Arti*,*cial Intelligence*, 29, 289}338.
PAYNE, S. J. (1991). Display-based action at the user interface. *International Journal of Man-Machine Studies*, 35, 275}289.
PIAGET, J. & INHELDER, B. (1969). ¹*he Psychology of the Child*. New York: Basic Books.
RICHARDS, B. L., KRAAN, & KUIPERS, B. J. (1992). Automatic abduction of qualitative models, In *Proceedings of the 10th National Conference on Arti*,*cial Intelligence*, AAAI, pp. 723}728.
VEENMAN, M. V. J., ELSHOUT, J. J. & BUSATO, V. V. (1994). Metacognitive mediation in learning with computer-based simulations. *Computers in Human Behaviour*, 10, 93}106.
WASHIO, T. &MOTODA, H. (1997). Discovering admissible models of complex systems based on scale-types and identity constraints. *Proceedings of the 15th International Joint Conference on Arti*,*cial Intelligence*, pp. 810}817. Japan.
WEINBERG, G. M. (1971). Learning and meta-learning using a black box. *Cybernetica*, XIV, 125}138.